EECS 22L: Software Engineering Project in C Language

Lecture 11

Rainer Dömer

doemer@uci.edu

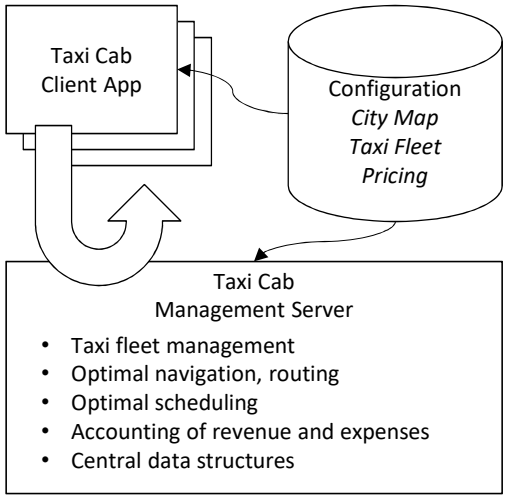
The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 11: Overview

- Project 2 Technical Discussion and Advise
  - Software architecture and components
  - Client-server communication
  - GUI programming
- Towards Object Oriented Programming in C++
  - Introduction to C++ concepts from the C perspective
  - Introduction to classes, objects and strings

Project 2: Software Architecture

- Overall System Specification (designed by consultant)

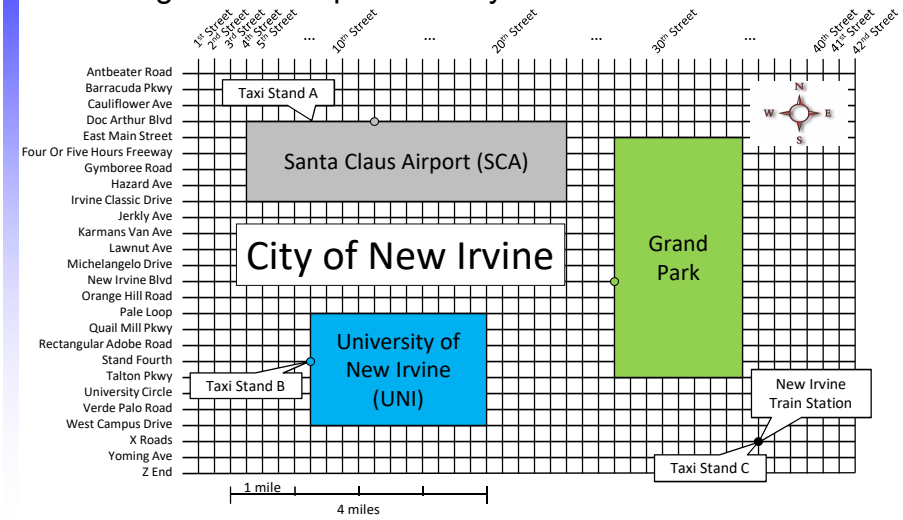


Project 2: Software Architecture

- Overall System Specification (designed by consultant)
  - Taxi Cab Management Server
    - Central server connected to the internet (with GUI dashboard)
    - Taxi fleet management
    - Optimal navigation, routing (and scheduling)
    - Accounting of revenue and expenses
    - Central data structures
  - Taxi Cab Client App
    - Customer interface to request taxi rides (with GUI interface)
    - Networked client app that communicates with the central server
  - Configuration Files
    - City Map (Grid, street names, landmark locations, etc.)
    - Taxi Fleet (Number of taxi cabs, capacity and parking locations, etc.)
    - Pricing Structure (Base price per hire, distance dependent rate, etc.)
    - ➢ All described in human and machine readable text file

Project 2: Software Configuration

- Configuration: Map of the City of New Irvine



Project 2: Software Configuration

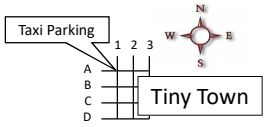
- Configuration: Map of the City of New Irvine
  - Configuration file **"NewIrvine.map"**

```
MAP City of New Irvine

GRID 26 42

STREET_NAMES_EAST_WEST Antbeater Road, Barracuda Pkwy, Cauliflower Ave, Doc Arthur
Blvd, East Main Street, Four Or Five Hours Freeway, Gymboree Road, Hazard Ave, Irvine
Classic Drive, Jerkly Ave, Karmans Van Ave, Lawnut Ave, Michelangelo Drive, New Irvine
Blvd, Orange Hill Road, Pale Loop, Quail Mill Pkwy, Rectangular Adobe Road, Stand
Fourth, Talton Pkwy, University Circle, Verde Palo Road, West Campus Drive, X Roads,
Yoming Ave, Z End

STREET_NAMES_NORTH_SOUTH 1st Street, 2nd Street, 3rd Street, 4th Street, 5th Street,
[…street names omitted for brevity…] 42nd Street

LANDMARK S8(P8,W19) University of New Irvine (UNI)
LANDMARK D4(D12,I24) Santa Claus Airport (SCA)
LANDMARK N27(E27,T35) Grand Park
LANDMARK X36 New Irvine Train Station

TAXI_STAND D8(12) Taxi Stand A
TAXI_STAND S8(12) Taxi Stand B
TAXI_STAND X36(12) Taxi Stand C
```

Project 2: Software Configuration

- Configuration: Map of Tiny Town (minimal example)
  - Configuration file **"TinyTown.map"**

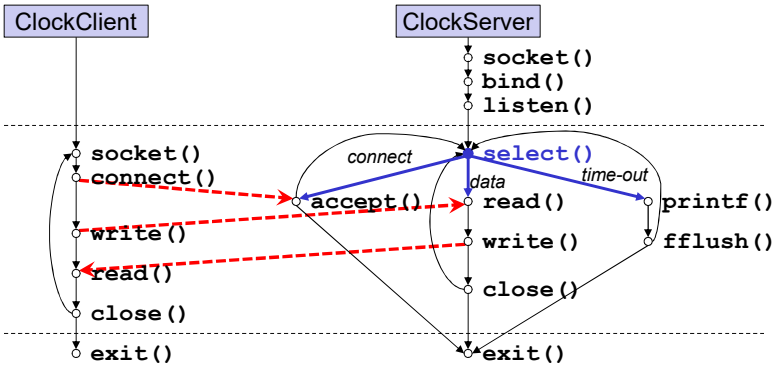
```
MAP Tiny Town
GRID 4 3
TAXI_STAND A1(3) Taxi Parking
```



  - ➢ No landmarks given, so assume default: no landmarks
    - ➢ Streets are reaching over the entire grid
  - ➢ No street names given, so use default names instead:
    - **STREET_NAMES_EAST_WEST A, B, C, D**
    - **STREET_NAMES_NORTH_SOUTH 1, 2, 3**

Project 2: Client-Server Communication

- Multiplexing multiple client connections with **select()**
  - ClockServer example: **~eecs22/ClockServer.tar.gz**



  - ➢ Wait simultaneously to connect, to transfer data, or for time-out!
  - ➢ Keep sequential execution short
  - ➢ Limit client-server interaction to one request at a time

Project 2: Client-Server Communication

- Communication Example: *"Call a Taxi from UNI to SCA"*
  - ➢ Reconsider: Handle one request at a time, keep sequences short!
  - – Client: `Hello Server!`
  - – Server: `ERROR invalid message "Hello Server!"`
  - – Client: `REQUEST_POSITION Taxi7`
  - – Server: `OK Taxi7 POSITION J2 ETA D4 11:45`
  - – Client: `REQUEST_TAXI S8 TO D4 ASAP`
  - – Server: `OK Taxi7 PICKUP S8 10:15 DROPOFF D4 11:42 $8.75`
    `CONFIRM #10042`
  - – Client: `CONFIRM #10042`
  - – Server: `OK Taxi7 POSITION D8 ETA S8 10:15`
  - ➢ *Unconfirmed requests should expire after some period (so that there is no problem when the client doesn't cancel the request)*

Decoupled requests by confirmation number

Project 2: Client-Server Communication

- Protocol Specification Example (Backus-Naur Form, BNF)
  - – Client Request

```
<request>  ::= REQUEST_TAXI <location> TO <location> <special>*
             |  REQUEST_POSITION <taxi>
             |  CONFIRM <reservation>
             |  CANCEL <reservation>
<location>::= <pos>
             |  CORNER <street_name> AND <street_name>
             |  <landmark_name>
<special>  ::= [ASAP]
             |  [AT <time>]
             |  [FOR <number_persons>]
             |  [NON_STOP]
<time>     ::= <hours>:<minutes>
```

  - – Server Response

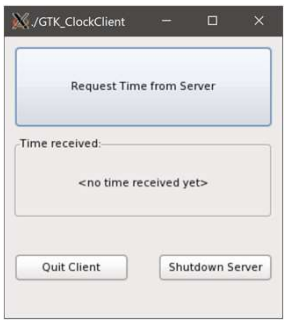
```
<response>::= OK <taxi> PICKUP <pos> [<time>] DROPOFF <pos> [<time>]
                  $<amount> CONFIRM <reservation>
             |  DECLINED <reason>
             |  OK <taxi> POSITION <position> [ETA <position> <time>]
             |  INVALID <reservation>
             |  ERROR <message>
<amount>   ::= <dollars>.<cents>
```

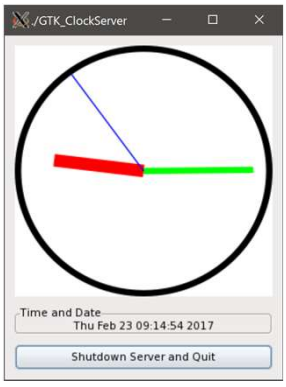
Project 2: GUI Programming

- GTK+ 2.0 Library Infrastructure
  - Tutorial: https://developer.gnome.org/gtk-tutorial/stable/
  - Reference: https://developer.gnome.org/gtk2/2.24/
  - Widgets: https://developer.gnome.org/gtk2/2.24/ch02.html
  - Cairo graphics: https://cairographics.org/samples/
  - Cairo tutorial: http://zetcode.com/gfx/cairo/
- GUI clock server example:
  - **`~eecs22/GTK_ClockServer.tar.gz`**
  - **`GTK_ClockClient.c`**
  - **`GTK_ClockServer.c`**
  - **`Makefile, README`**

Project 2: GTK Clock Server Example

- Screenshot of the Client and Server Windows





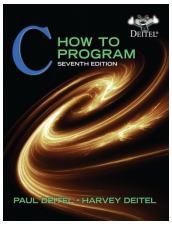
Object Oriented Programming

- Towards Object Oriented Programming in C++
  - C++ can be seen as “incremented” or “improved” C
  - C++ offers a number of new features, including:
    - Inline functions
    - References
    - Default arguments
    - Function and operator overloading
    - Classes and objects
    - Member functions (methods)
    - Constructor and destructor
    - Class and function templates
    - Class inheritance
    - Polymorphism
    - Exception handling

Object Oriented Programming

- Brief Introduction to C++
  - Selected slides from supplemental text book:

    Paul Deitel, Harvey Deitel,
    “C: How to Program”,
    Seventh Edition,
    Prentice Hall, 2013.



  - Excerpts from Chapter 16:
    Introduction to Classes, Objects and Strings